Path Tracer - Documentation

By Kevin Buman

Setup

1. Rename `CMakeListsCopy.txt` to `CMakeLists.txt`.
2. Replace all `<path_to>` in `CMakeLists.txt`.
3. Generate your project files.
4. In order to run the program you have to set the working directory (i.e. where the program is run from) to the root directory.

Developed with Mingw on Windows.

Libraries used:

- GLM
- SDL2

Configuration Options

```
// main.cc
BasicRenderer<sRGB, CustomResolution> renderer({480,480}); // Basic Renderer
(single intersection) with diffuse colors only
MonteCarloRenderer<sRGB, CustomResolution> renderer({480,480}); // Renderer in
sRGB Mode with a custom resolution of 480x480
MonteCarloRenderer<sRGB, HD> renderer; // Renderer in sRGB Mode with a
resolution of 1280x720
renderer.SetSampling(1);            // Set sampling quality (1 = render every
pixel, 2 = render 2x2 grids, etc.)
renderer.SetRenderQuality(4096);    // Set rays per pixel
renderer.SetAASigma(0.5f);          // Set sigma of the normal distribution,
default 0.5f
renderer.SetAA(true);               // enable anti aliasing
```

Lab 01

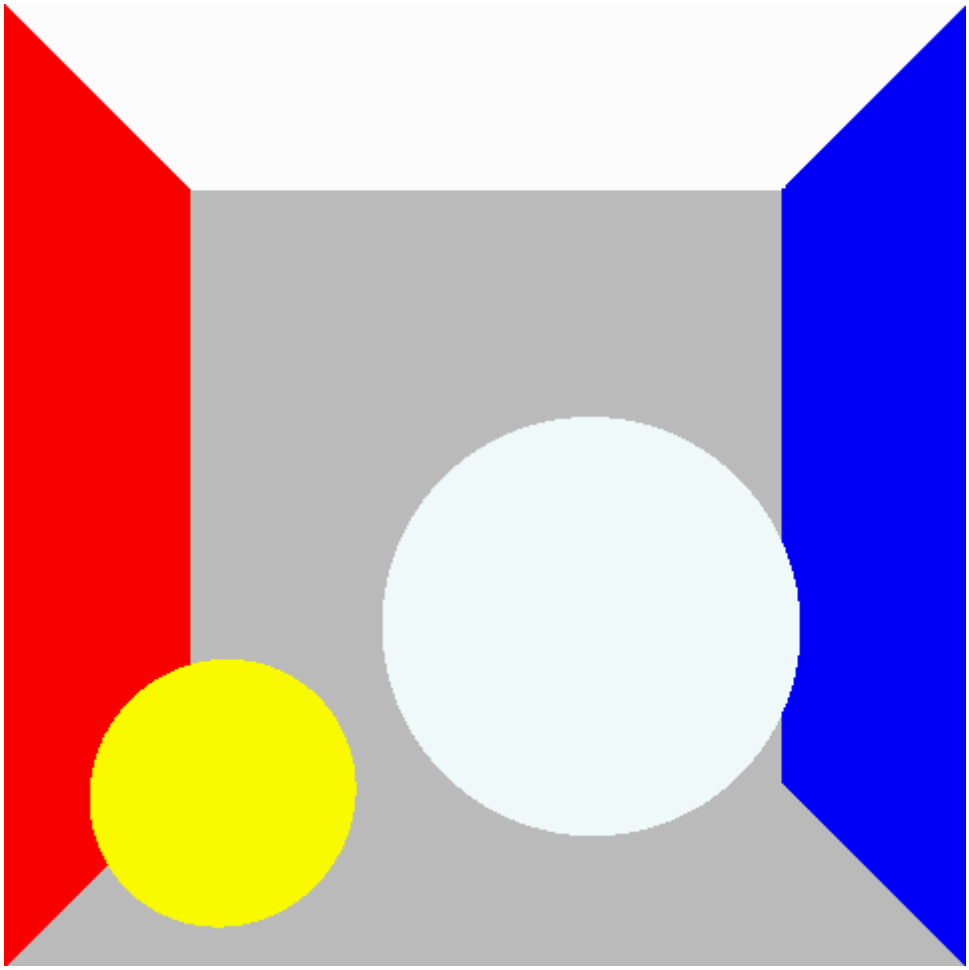
- Two color modes, which are applied at the end of the render process. Before that, I only work on floating point rgb

```
void RGB::Process(glm::vec3 &base_color) const {
  base_color[0] = glm::clamp(base_color[0], 0.0f, 1.0f) * 255;
  base_color[1] = glm::clamp(base_color[1], 0.0f, 1.0f) * 255;
  base_color[2] = glm::clamp(base_color[2], 0.0f, 1.0f) * 255;
}

void sRGB::Process(glm::vec3 &base_color) const {
  base_color[0] = glm::pow(glm::clamp(base_color[0], 0.0f, 1.0f), (1 / m_gamma))
* 255;
  base_color[1] = glm::pow(glm::clamp(base_color[1], 0.0f, 1.0f), (1 / m_gamma))
* 255;
  base_color[2] = glm::pow(glm::clamp(base_color[2], 0.0f, 1.0f), (1 / m_gamma))
* 255;
}
```

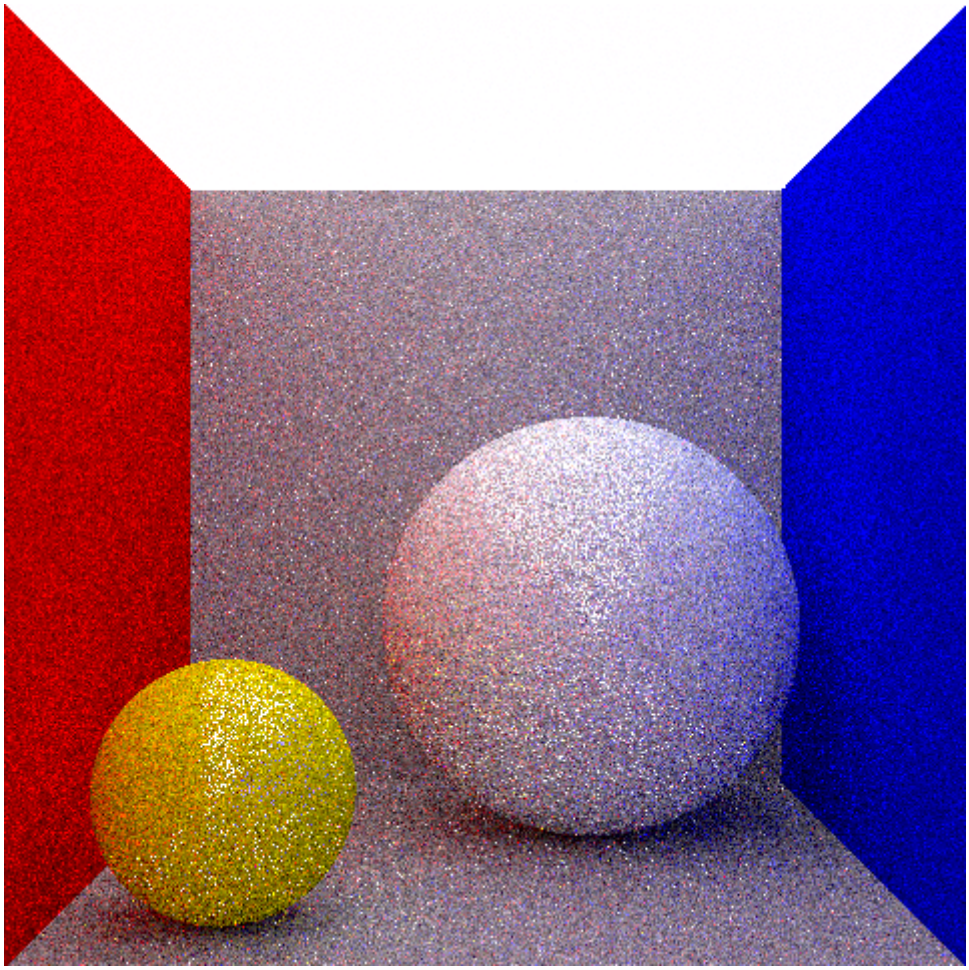
Screenshots

Lab 02

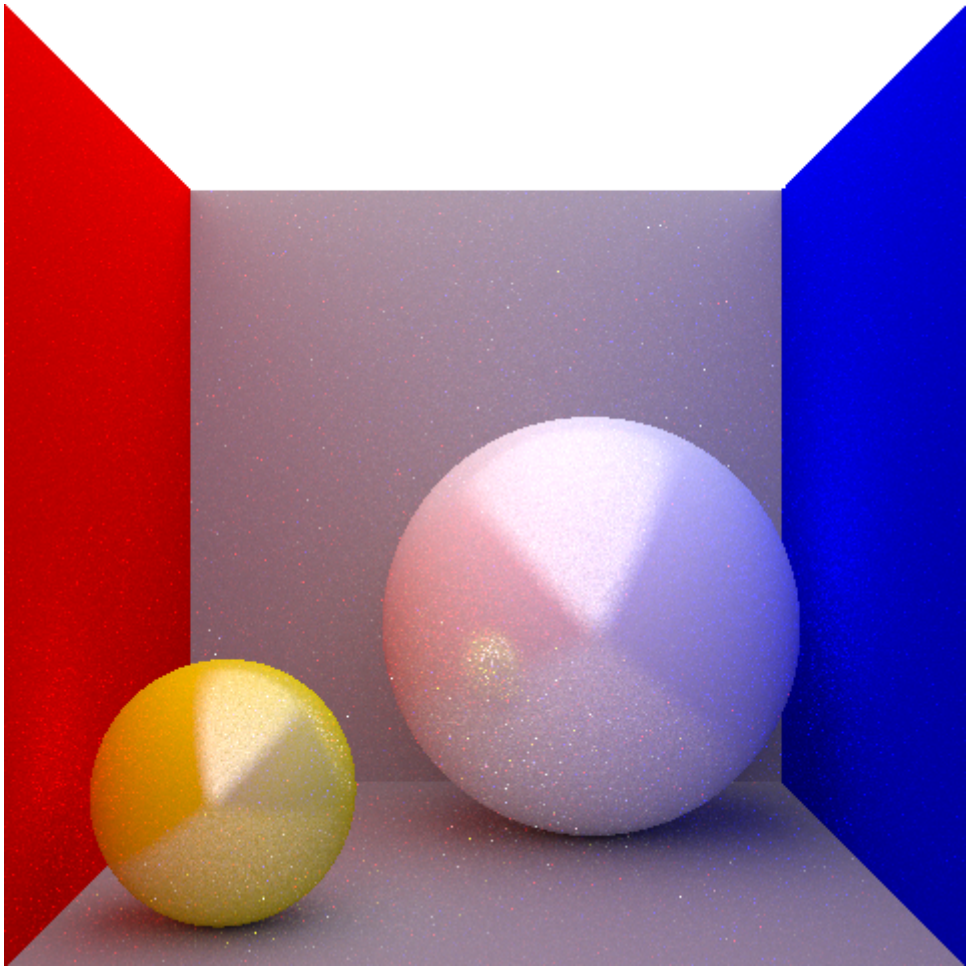


- Basic path tracer with diffuse and emmissive colors

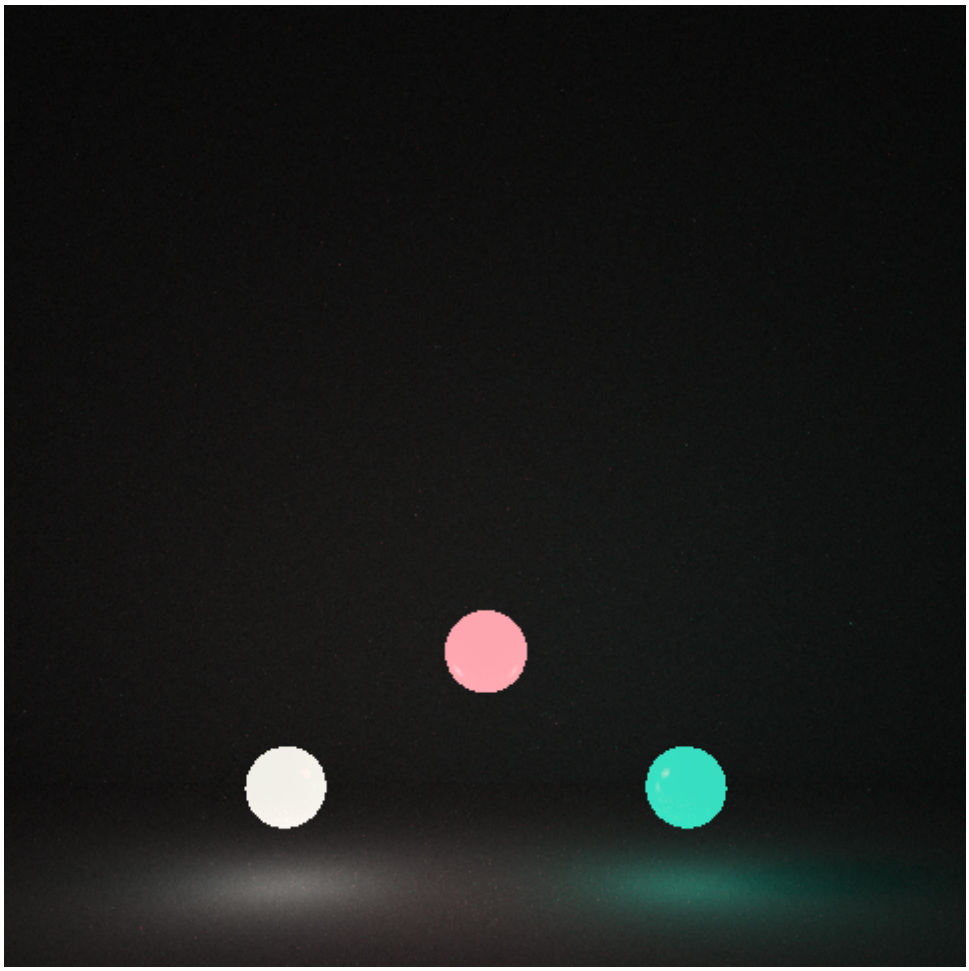
Lab 03



- Reflections with 32 rays per pixels

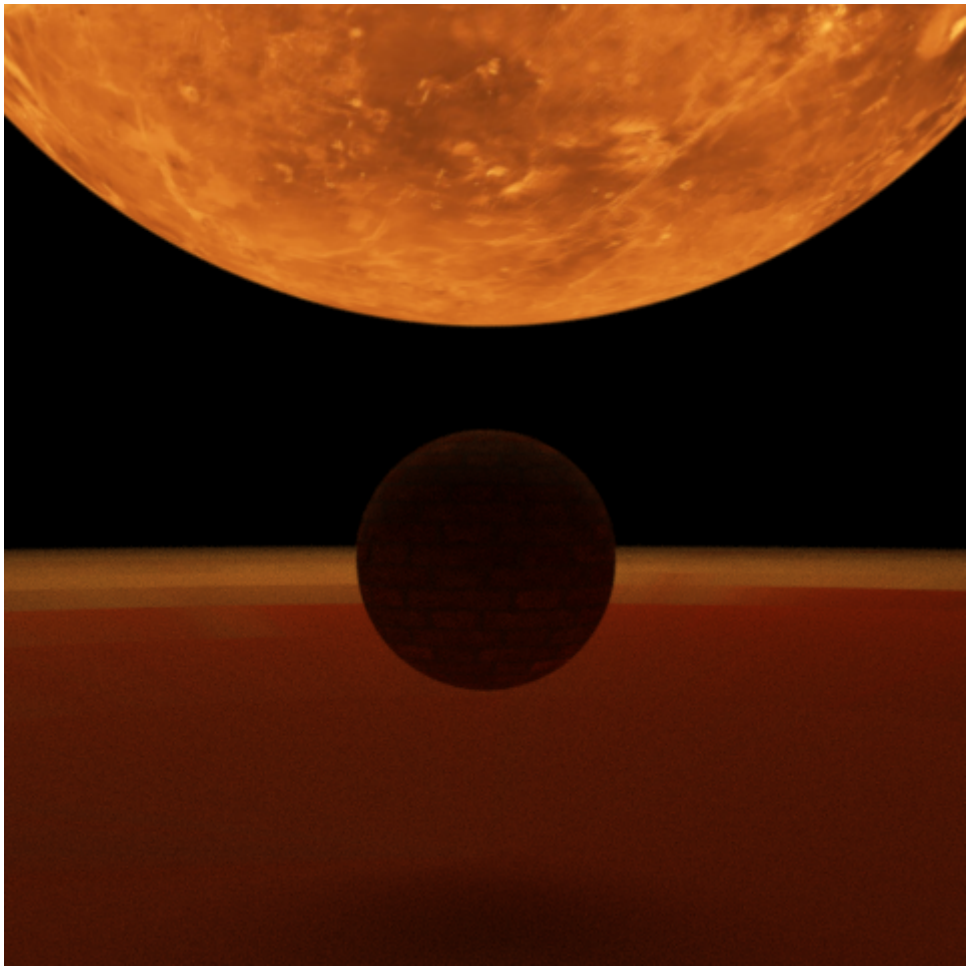


- 8192 rays per pixel

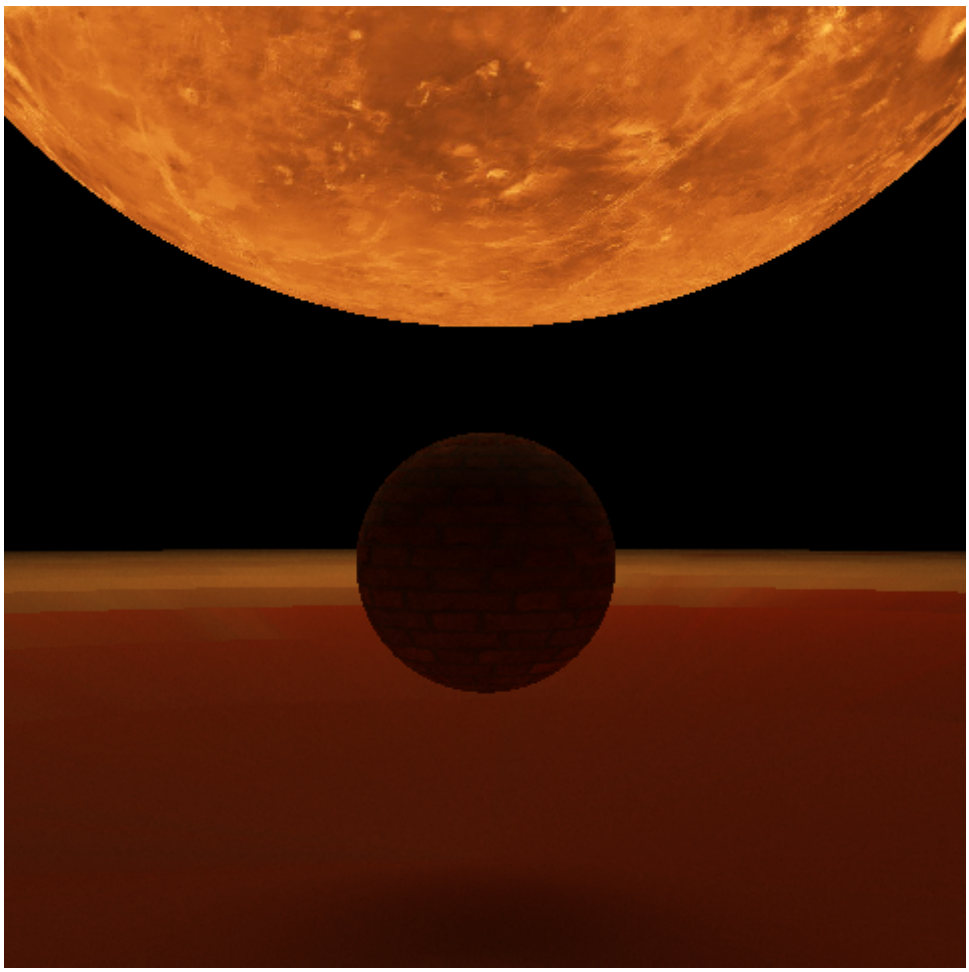


- Additional Custom Scene with 3 Emissive spheres(two on the side are also specular in the same color as the middle one) - 4096 rays per pixel

Lab 04

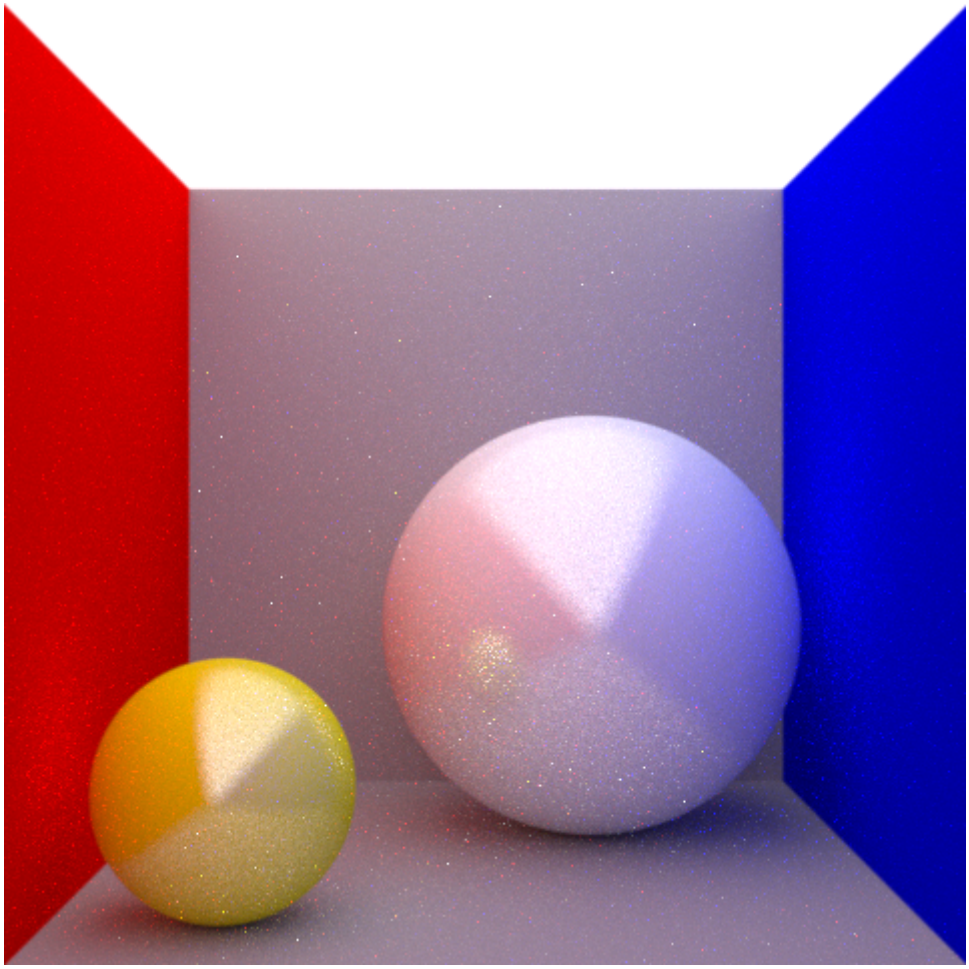


- Custom scene with 1024 rays per pixel

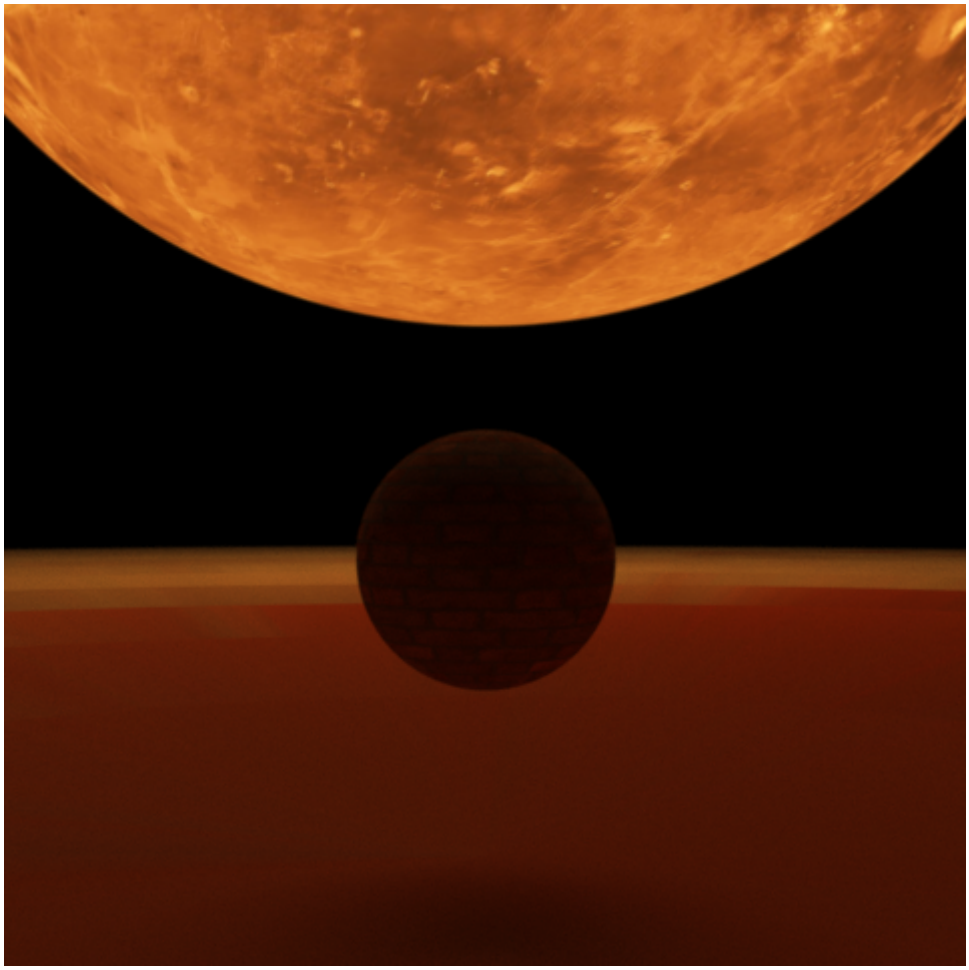


- Custom Scene with 4096 rays per pixel and bitmap textures

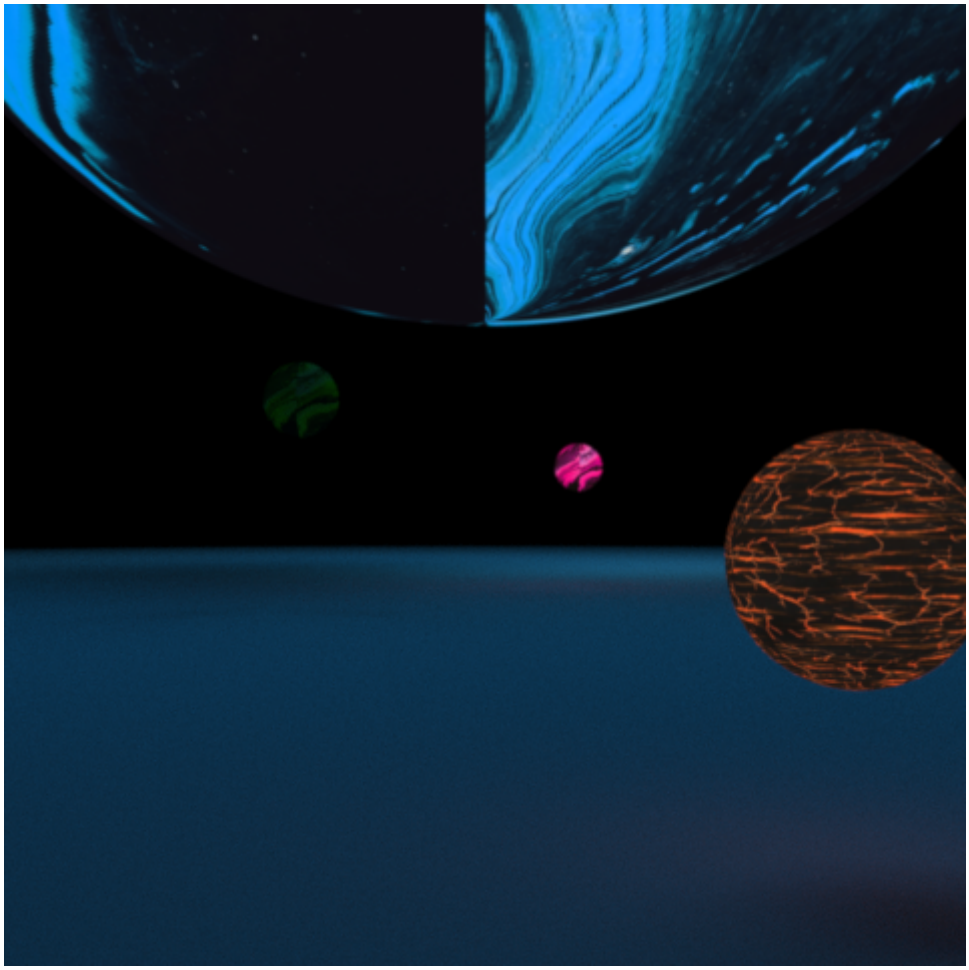
Lab 05



- Cornell Box with 8192 rays per pixel and gaussian anti aliasing (sigma = 0.5f)



- Custom Scene 1 with 4096 rays per pixel and anti aliasing turned on(sigma = 0.5f)



- Custom Scene 2 with 4096 rays per pixel and anti aliasing turned on (sigma = 0.5f)

Additional Screenshots

- See `renders/`